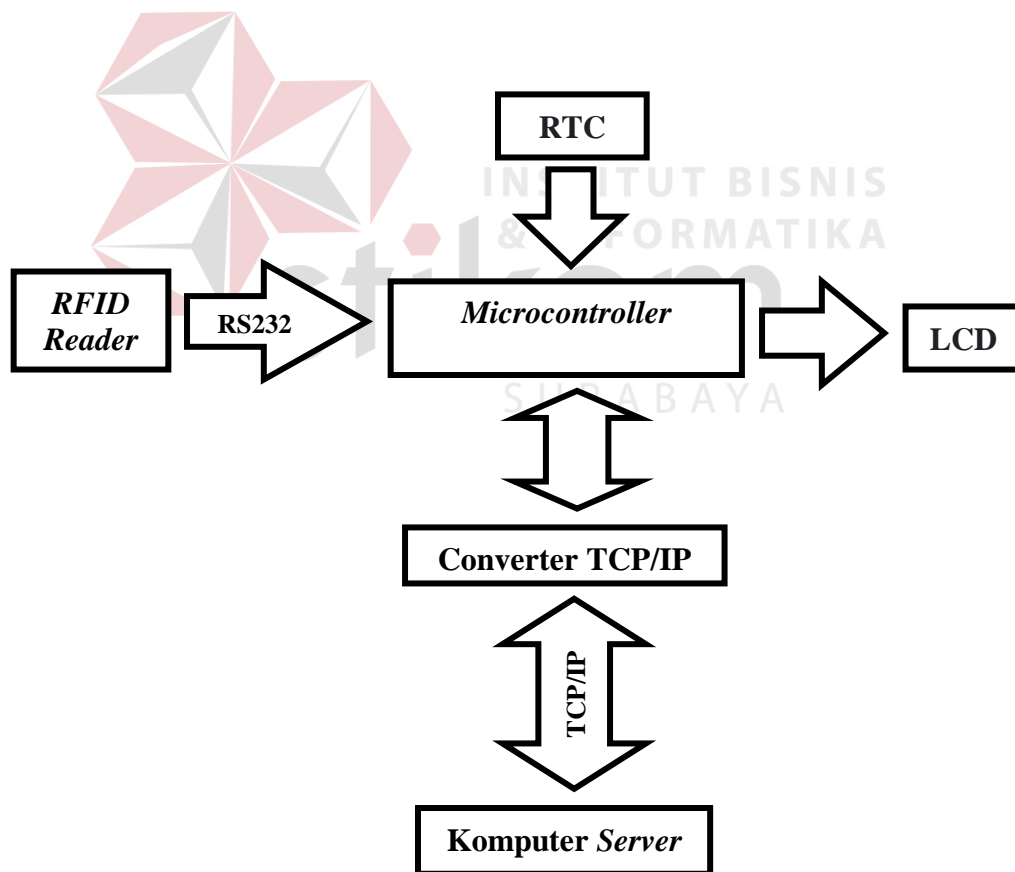


BAB III

METODE PENELITIAN

3.1. Model Penelitian

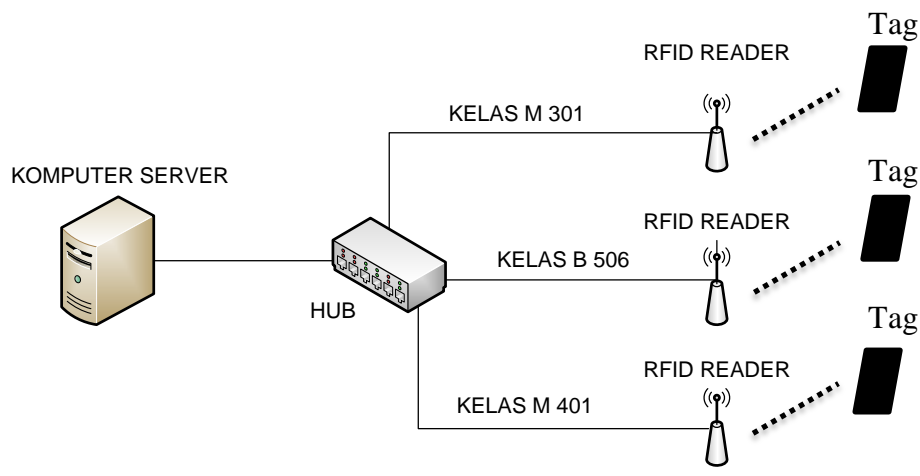
Pengerjaan Tugas Akhir ini dapat terlihat jelas dari blok diagram yang tampak pada Gambar 3.1. Blok diagram tersebut menggambarkan proses RFID mengirim data dari kartu sampai ke komputer *server* hingga data tersebut dikembalikan ke *microcontroller* dan ditampilkan di LCD Terdapat beberapa komponen penting pada blok diagram tersebut antara lain adalah modul RFID Starter Kit, modul RTC, modul WIZ110SR, dan *microcontroller*.



Gambar 3.1. Blok Diagram

- a. RFID menerima data dari kartu kemudian data dikirimkan ke *microcontroller* menggunakan komunikasi serial (RS232).
- b. Data ID dari kartu kemudian dirubah menjadi data NIM melalui *microcontroller*.
- c. Data dari *microcontroller* dikirim ke PC *server* melalui konverter TCP/IP.
- d. Setelah data diterima oleh PC *server* maka data dibandingkan dengan database waktu kedatangan mahasiswa dengan waktu yang ditetapkan sesuai jadwal.
- e. Jika waktu kedatangan kurang dari waktu yang ditetapkan jadwal, maka mahasiswa dinyatakan dengan keterangan tepat. Jika waktu lebih dari waktu yang ditetapkan jadwal, maka mahasiswa dinyatakan dengan keterangan terlambat.
- f. Setelah data dibandingkan maka data dikirimkan kembali ke *microcontroller*. Data yang dikirimkan berisi NIM, keterangan keterlambatan, dan jam *display*.
- g. Data diterima oleh *microcontroller* kemudian di tampilkan di LCD.

3.2. Cara Kerja Sistem Secara Keseluruhan



Gambar 3.2. Sistem Keseluruhan

Dari Gambar 3.2 dapat diketahui bahwa alat absensi (*RFID reader*) diletakkan di setiap kelas yaitu pada kelas M 301, B 506, dan M 401. Seluruh masing-masing *tag* mengirimkan data ID ke *RFID reader*. *RFID reader* setelah menerima data dari *tag* maka data ID tersebut diterima oleh *microcontroller* melalui komunikasi serial (RS-232). Data ID dari *microcontroller* kemudian dikirimkan ke PC server melalui komunikasi TCP/IP. Sebelum melewati PC server pengiriman data ID dari ketiga kelas tersebut melalui *hub* karena lebih dari dua alat absensi. Pengiriman data tersebut dilakukan secara berurutan karena ada waktu jeda 2 detik.

Data ID setelah diterima oleh PC server maka data ID akan dirubah menjadi data NIM dan ditambahkan keterangan keterlambatan. Pada PC server aplikasi yang digunakan menggunakan Visual Basic 6 dan database *MySql*. Pengolahan data ID dirubah menjadi NIM yaitu dengan mendaftarkan ID dan NIM pada database *MySql*. Setelah data dibandingkan maka data tersebut dikirimkan

kembali ke *microcontroller*. Data NIM dan keterlambatan sudah diterima oleh *microcontroller* dan ditambahkan catatan waktu pada RTC kemudian ditampilkan di *display* LCD.

3.3. Perancangan Perangkat Keras

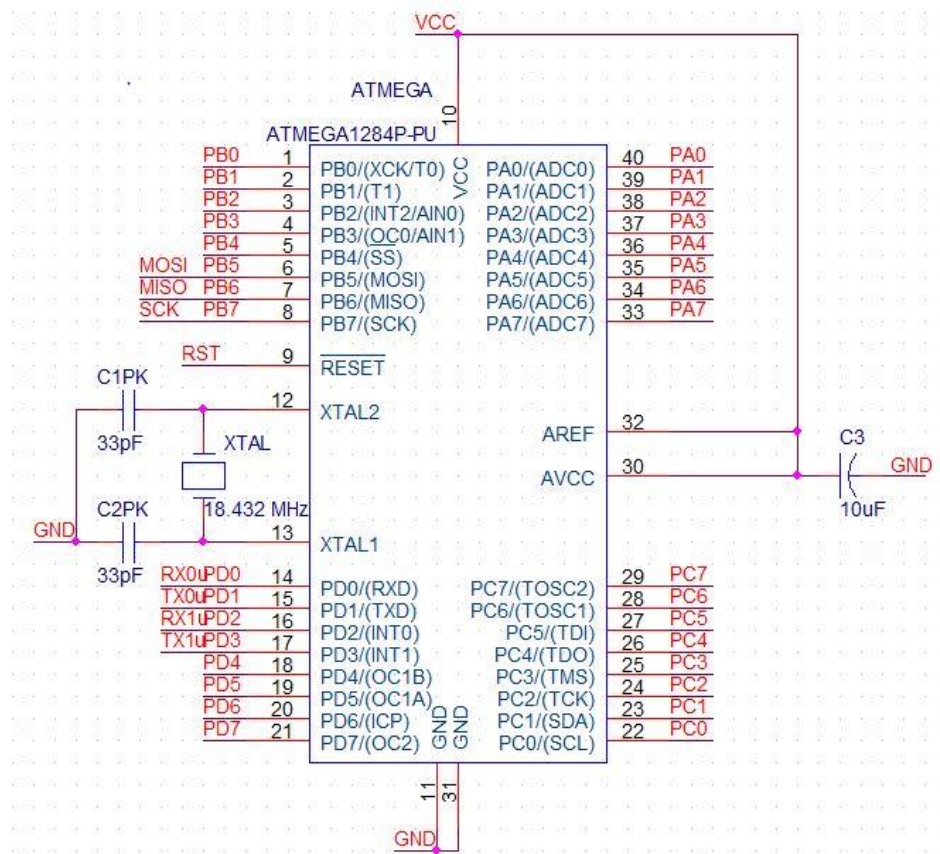
Hardware merupakan perangkat keras dalam sistem komputer yang secara fisik terlihat dan dapat disentuh. Perangkat keras yang dibutuhkan guna mendukung kelancaran program antara lain terdiri dari modul RFID Starter Kit, modul *RTC*, rangkaian LCD, rangkaian max232, modul WIZ110SR serta penggunaan rangkaian sistem minimum.

3.3.1. Rangkaian *Microcontroller*

Pada proyek tugas akhir ini dibuat piranti pengendali menggunakan *microcontroller* keluaran AVR, yaitu ATMEGA1284P-PU. Pengendalian program pada *microcontroller* ini diperlukan rangkaian sistem minimum. Rangkaian sistem minimum adalah rangkaian elektronika yang terdiri dari komponen-komponen dasar yang dibutuhkan oleh suatu *microcontroller* dapat berfungsi dengan baik.

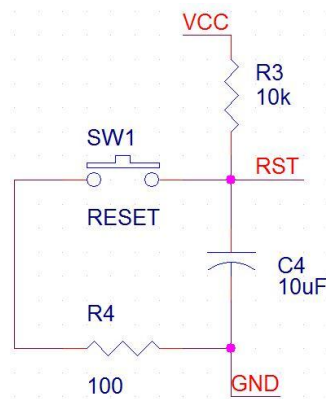
A. Rangkaian sistem minimum *microcontroller*

Rangkaian sistem minimum terbagi menjadi 3 rangkaian utama yaitu rangkaian IC, rangkaian *reset* dan rangkaian *crystal*. Rangkaian IC sistem minimum dapat lihat pada Gambar 3.3.

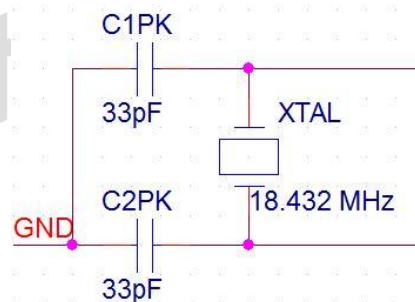


Gambar 3.3. Rangkaian Sistem Minimum ATMEGA1284P-PU

Pada Gambar 3.3 menjelaskan bahwa rangkaian pada *microcontroller* tipe ATMEGA1284P-PU terdapat 40 pin. (Atmel, 2009). IC ini memiliki 4 Port. Port A digunakan untuk *output* LCD. Port B digunakan untuk pengiriman program ke komputer menggunakan *downloader*. Port C digunakan untuk *input* RTC. Port D digunakan untuk *output* data ke komputer. Tegangan masukan DC 5 Volt diparalel dengan kapasitor 100 uF sebagai filter supaya tidak ada kekacauan data. Sedangkan rangkaian *reset* dapat dilihat pada Gambar 3.4.

Gambar 3.4. Rangkaian *Reset*

Pada Gambar 3.4 menjelaskan bahwa rangkaian *reset* pada tombol *reset* (SW1) digunakan untuk mengurangi *noise* serta memiliki fungsi terpenting yaitu untuk melakukan *reset* saat pertama kali catu daya dinyalakan. *Reset* untuk pertama kali merupakan hal yang terpenting sehingga dapat memastikan bahwa program telah berada pada posisi awal. (Wahyuni, 2013)

Gambar 3.5. Rangkaian *Crystal*

Pada Gambar 3.5 menjelaskan bahwa rangkaian *crystal* yang digunakan sebesar 18,432 MHz karena sistem minimum dapat berjalan normal diatas *crystal* 14 MHz. Penggunaan *crystal* 18,432 MHz dapat diperoleh dengan perhitungan UBRR karena mempunyai nilai *error* komunikasi serial 0 persen. Perhitungan nilai *error* dapat diperoleh sebagai berikut :

1. Perhitungan UBRR

Baudrate yang digunakan pada Tugas Akhir ini adalah 9600 bps sedangkan frekuensi osilator yang digunakan sebesar 18,432 MHz. Berikut ini perhitungan UBRR.

$$UBRR = \frac{f_{osc}}{16.Boudrate} - 1 \dots\dots\dots (3.1)$$

$$UBRR = \frac{18432000}{16.9600} - 1 \dots\dots\dots (3.2)$$

$$UBRR = \frac{18432000}{153600} - 1 \dots\dots\dots (3.3)$$

$$UBRR = 119 \dots\dots\dots (3.4)$$

2. Perhitungan BAUD

Dari perhitungan UBRR didapatkan nilai 119 maka untuk pembuktian perhitungan *baudrate* pada Tugas Akhir ini sebagai berikut.

$$BOUD = \frac{f_{osc}}{(16.(UBRR+1))} \dots\dots\dots (3.5)$$

$$BOUD = \frac{18432000}{(16.(119+1))} \dots\dots\dots (3.6)$$

$$BOUD = \frac{18432000}{(16.(120))} \dots\dots\dots (3.7)$$

$$BOUD = \frac{18432000}{(16.(120))} \dots\dots\dots (3.8)$$

$$BOUD = \frac{18432000}{1920} \dots\dots\dots (3.9)$$

$$BOUD = 9600 \text{ bps} \dots\dots\dots (3.10)$$

3. Perhitungan 0 % Error

Perhitungan nilai eror dengan *baudrate* 9600 bps dan UBRR 119 adalah :

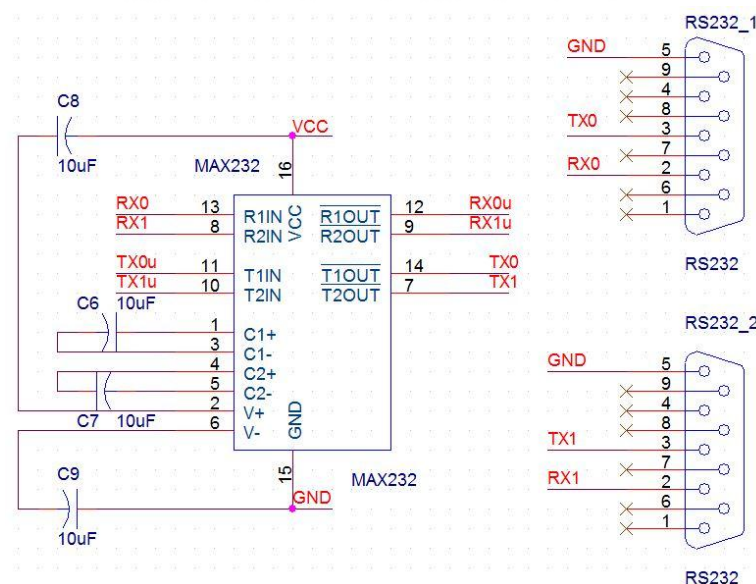
$$\% Error = \frac{BAUD_{Actual} - BAUD_{Awal}}{BAUD_{Awal}} \times 100\% \dots\dots\dots (3.11)$$

$$\% Error = \frac{9600 - 9600}{9600} \times 100\% \dots\dots\dots (3.12)$$

$$\% Error = 0 \% \dots\dots\dots (3.13)$$

3.3.2. Rangkaian Max232

Rangkaian serial merupakan rangkaian yang dibutuhkan agar sebuah *microcontroller* dapat berkomunikasi secara serial dengan komputer atau peralatan lain. *Microcontroller* menggunakan TTL sebagai *input* dan *output* data, yang berbeda dengan komputer personal. Oleh sebab itu, dibutuhkan sebuah rangkaian yang dapat digunakan untuk menjebatani hal tersebut. Pada Gambar 3.6 terlihat rangkaian serial dengan menggunakan IC MAX232.



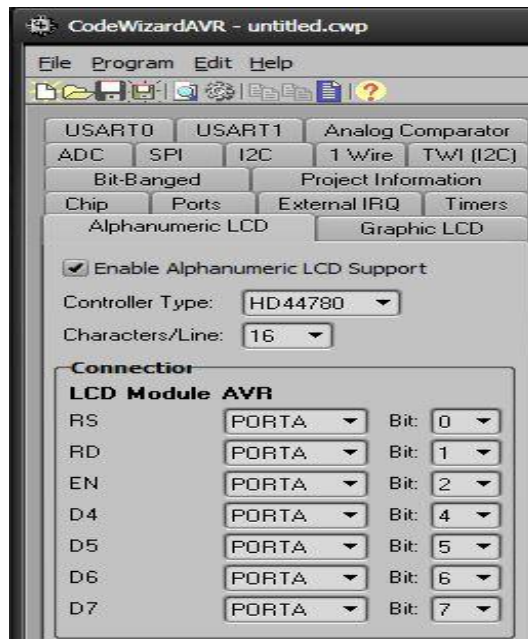
Gambar 3.6. Rangkaian Max232

Cara kerja rangkaian pada Gambar 3.6 terdapat 2 buah serial, keduanya terlihat bahwa antara RX0u dan TX0u dibuat *cross* (terbalik), T1in masuk pada TXD *microcontroller* dan R1in masuk pada DB9 (*interface serial* ke komputer) serta T1OUT masuk pada DB9 dan R1OUT Menuju RXD *microcontroller*.

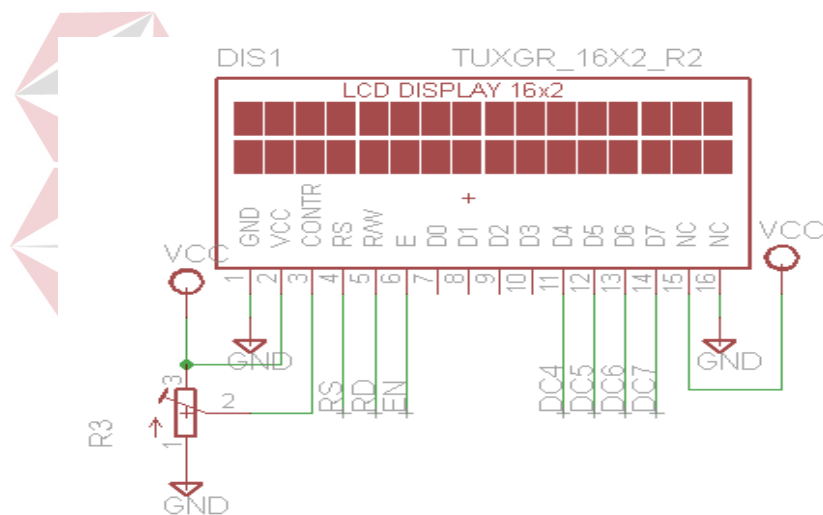
Terdapat 4 buah kapasitor yang digunakan untuk memperhalus data yang masuk serta mengurangi *noise*, dimana nilai-nilai yang ditentukan disesuaikan dengan *datasheet* pada IC MAX232. (Putra, 2011)

3.3.3. Rangkaian LCD

Perancangan rangkaian LCD menggunakan tipe 2x16 lebar *display* 2 baris 16 kolom, yang mempunyai 16 pin konektor. Pengaturan menggunakan CodeWizardAVR dapat dengan mudah mendefinisikan *port* yang terhubung dengan LCD. Pada Gambar 3.6 Port A yang digunakan hanya berjumlah 7 pin sedangkan pin 3 dibiarkan kosong, sedangkan pada LCD pin data yang digunakan sebanyak 4 pin sehingga pin data yang digunakan dari data pin 11-14. Pada penggunaan *microcontroller* AVR konfigurasi data bit khususnya pada ATMEGA1284P-PU hanya memerlukan 4 bit data untuk terkoneksi dengan *port* yang terhubung ke LCD dapat dilihat pada Gambar 3.7. (Darmawan, 2013). Sedangkan untuk *schematic* hubungan PORT C dengan LCD dapat dilihat pada Gambar 3.8.



Gambar 3.7. Konfigurasi Port A ke LCD



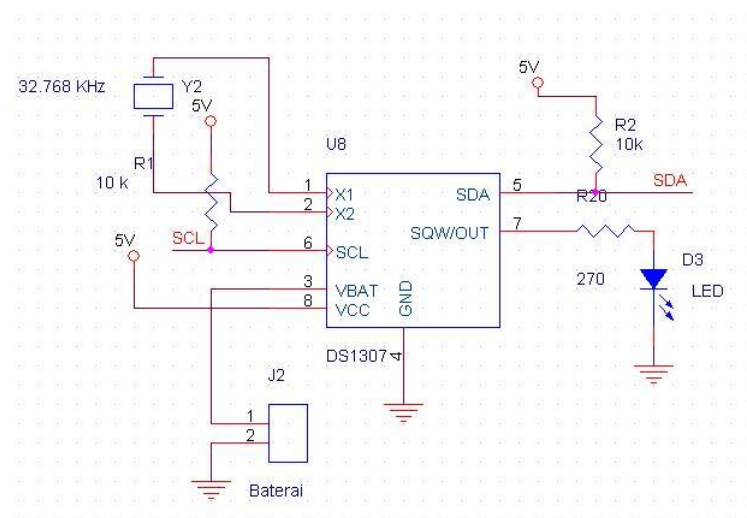
Gambar 3.8. Schematic Konfigurasi Port A dengan LCD

Program untuk menampilkan karakter LCD adalah sebagai berikut :

```
#include <mega1284p.h>
// Alphanumeric LCD functions
#include <alcd.h>
lcd_init(16);
lcd_gotoxy(3,0);
lcd_puts("WELCOME TO"); //menampilkan pada baris 0 dari kolom 3
lcd_gotoxy(0,1);
lcd_puts("STIKOM SURABAYA"); //menampilkan pada baris 1 dari
kolom 0
```

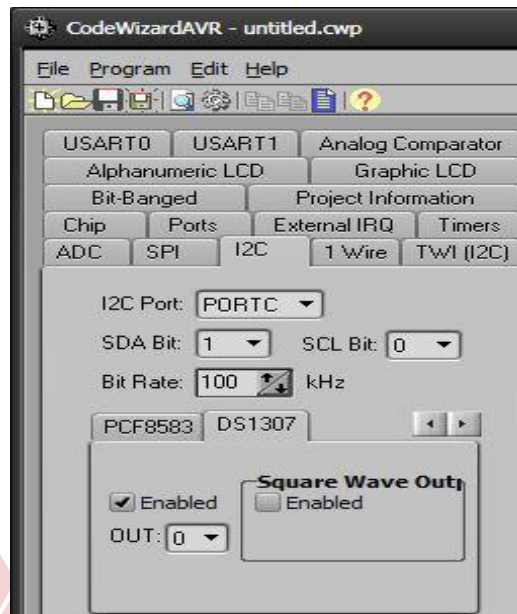
3.3.4. Rangkaian Modul RTC

Real Time Clock (RTC) merupakan suatu *chip* yang memiliki fungsi sebagai kalender dan jam elektronik di mana perhitungan hari, tanggal, bulan, tahun, jam, menit dan detik tersimpan di memori dengan alamat-alamat tertentu, dan *alarm* yang dapat di program keaktifannya. Sistem perhitungan jam dan kalender pada RTC berjalan secara otomatis dan berlanjut walaupun catu daya dimatikan. Pada pemakaiannya, RTC biasa dihubungkan ke *microcontroller*, dimana *microcontroller* tersebut hanya perlu melakukan pengaturan *mode* RTC, pengaturan waktu maupun pembacaan waktu saja. Komunikasi data pada IC DS1307 adalah *Inter Integrated Circuit* (I2C) yang membutuhkan kaki SDA & SCL untuk proses transfer data. RTC DS1307 menggunakan sebuah kristal 32,768Hz untuk *clock* sekaligus memerlukan baterai eksternal 3 volt yang terhubung ke pin Vbat dan *ground*. Pin X1 dan X2 dihubungkan dengan kristal osilator 32,768 KHz. Sedangkan pin SCL, SDA, dan SQW/OUT *dipull-up* dengan resistor (nilainya 1k sampai dengan 10k) ke vcc. Rangkaian RTC ini dapat dilihat pada Gambar 3.9.



Gambar 3.9. Rangkaian Modul RTC

RTC DS1307 dikonfigurasi dengan menggunakan fasilitas *CodeWizard* AVR. Konfigurasi RTC dengan *CodeWizard* dapat dilihat pada Gambar 3.10.



Gambar 3.10. Konfigurasi RTC dan CodeWizardAVR

Program untuk mengaktifkan RTC adalah sebagai berikut :

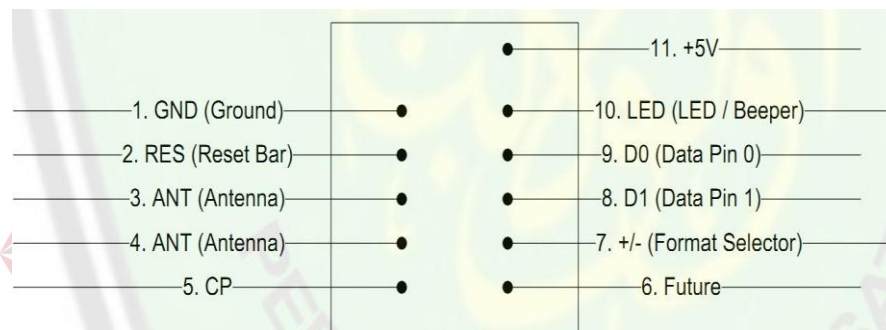
```
#include <mega1284p.h>
#include <i2c.h>
#include <ds1307.h>
lcd_init(20);
rtc_set_time(14,10,0);
rtc_set_date(5,18,5,12);

while (1)
{
    rtc_get_date(&hari,&tanggal,&bulan,&tahun);
    rtc_get_time(&jam,&menit,&detik);
}
```

3.3.5. Rangkaian Modul RFID Starter Kit

RFID Starter Kit merupakan suatu sarana pengembangan RFID berbasis *reader* tipe ID-12 yang telah dilengkapi dengan jalur komunikasi RS-232 serta indikator *buzzer* dan LED. Modul ini dapat digunakan dalam aplikasi mesin absensi RFID, RFID *access controller*, dan sebagainya.

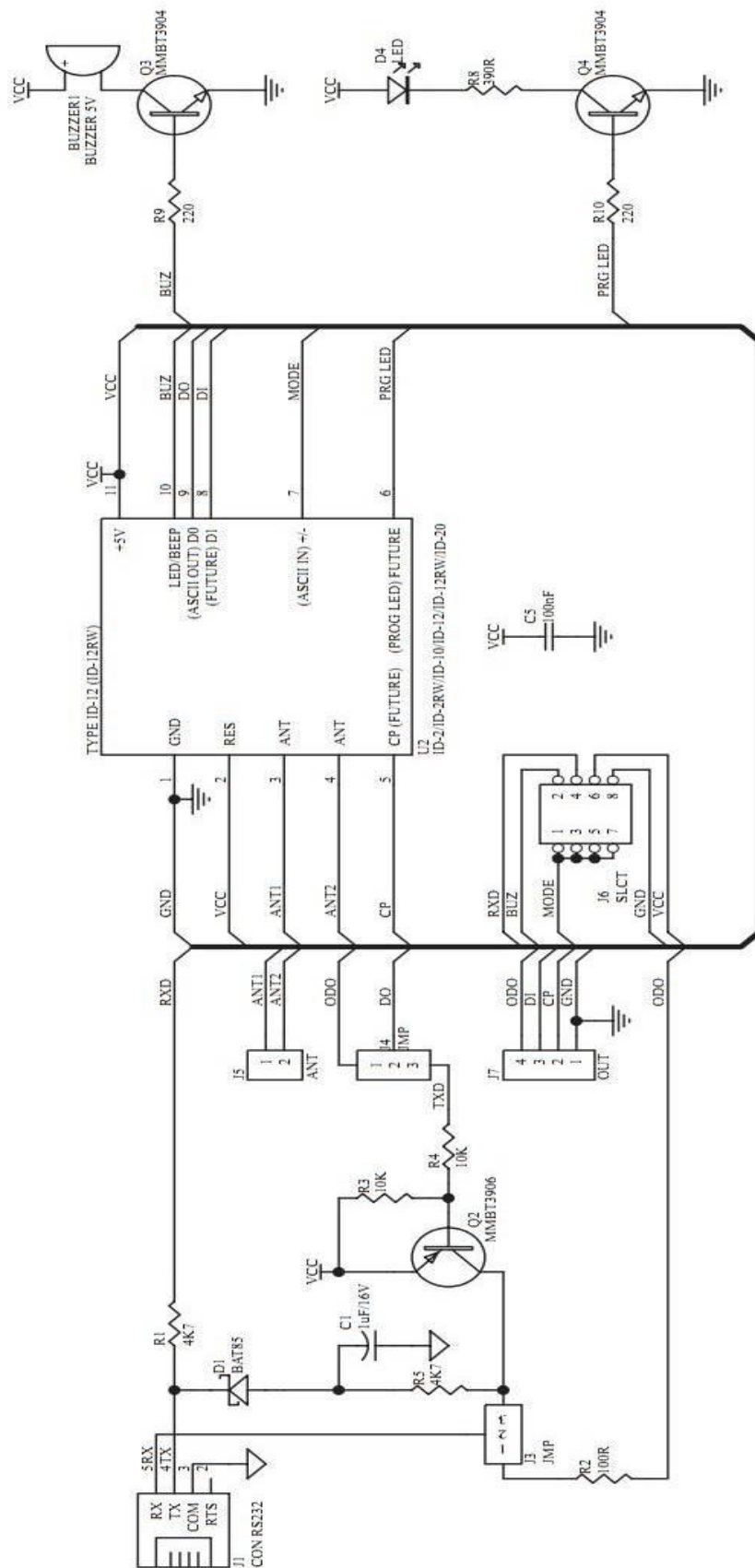
Reader RFID digunakan untuk membaca *tag* yang mendekati *reader* dalam jangkauan pembacaan. Dalam rangkaian ini, *reader* yang digunakan ID-12 sesuai *tag* RFID yang digunakan. ID-12 mempunyai spesifikasi tegangan catu daya 5V, jarak pembacaan 5-12 cm, dan frekuensi kerja 125 KHz. Rangkaian modul RFID Starter Kit dapat dilihat pada Gambar 3.12 dan keterangan IC ID-12 yang digunakan untuk aplikasi absensi yang akan digunakan dapat dilihat pada Gambar 3.11.



Gambar 3.11. Keterangan IC ID-12

Tabel 3.1. Deskripsi Pin Dan *Format Output Data*

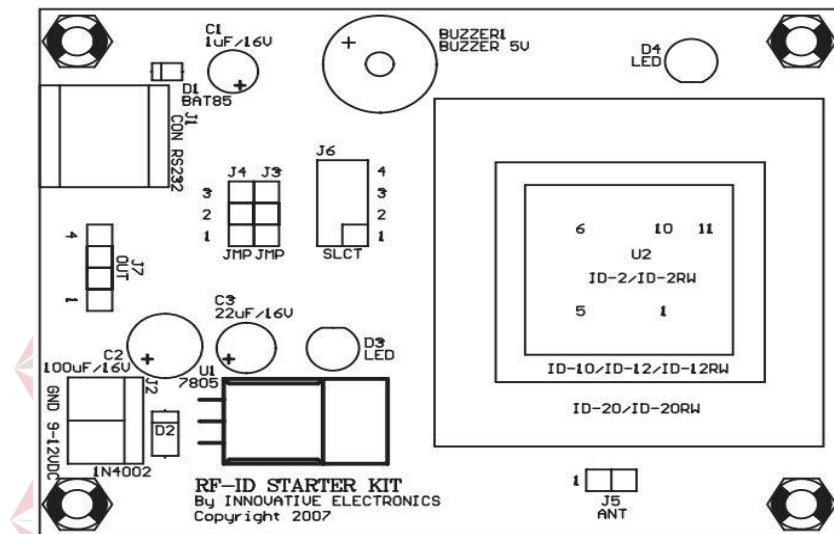
Pin	Description	ASCII	Magnet Emulation	Wiegand26
1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V	GND 0V
2	(Mengikat) Strap to +5V	Reset Bar	Reset Bar	Reset Bar
3	To external antenna and tuning capasitor	Antenna	Antenna	Antenna
4	To external antenna	Antenna	Antenna	Antenna
5	Card Present	No Function	Card Present *	No Function
6	Future	Future	Future	Future
7	<i>Format Selector</i> (+/-)	Strap to GND	Strap to Pin 10	Strap to +5V
8	Data 1	CMOS	Clock *	One Output *
9	Data 0	TTL data (inverted)	Data *	Zero output *
10	3.1 KHz Logic	Beeper/LeD	Beeper/LeD	Beeper/LeD



Gambar 3.12 Rangkaian modul RFID Starter Kit

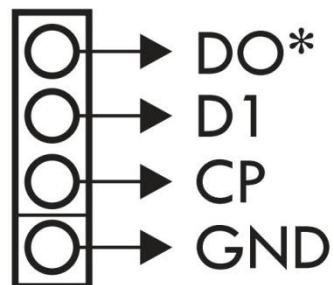
Tabel 3.2. Karakteristik Fisik Dan Operasional

Parameters	ID-12
Read Range	12+ cm
Dimension	26 mm x 25 mm x 7 mm
Frequency	EM4001 or compatible
Encoding	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 30mA nominal
I/O Output Current	-
Voltage Supply Range	+4.6V through +5.4V



Gambar 3.13. Tata Letak Komponen

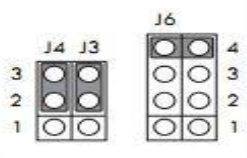
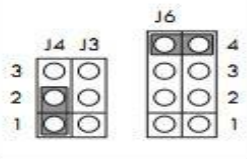
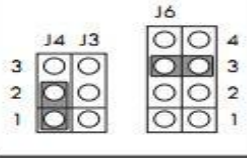
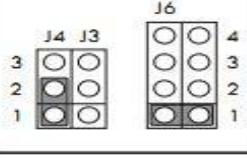
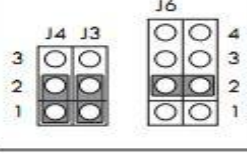
Pada Gambar 3.13 menjelaskan bahwa antenna dapat dihubungkan ke J5 untuk RFID *reader only* atau *reader/writer* yang memerlukan antenna eksternal, seperti ID-2 dan ID-2RW. Alokasi pin J7 dapat dilihat pada Gambar 3.14.



Gambar 3.14. Alokasi Pin J7

J7 hanya digunakan untuk RFID *reader only* dalam *mode* UART TTL (ASCII), Wiegand26, dan Magnet Emulation. J7 tidak boleh digunakan pada *mode* lain.

Pengaturan *jumper* J3, J4, dan J6 harus disesuaikan dengan jenis RFID (*reader only* atau *reader/writer*) serta *format* data RFID *reader* yang akan digunakan. Pengaturan *jumper* terdapat pada Gambar 3.15.

	RFID reader only dengan format data UART RS-232 (ASCII).
	RFID reader only dengan format data UART TTL (ASCII).
	RFID reader only dengan format data Wiegand26
	RFID reader only dengan format data Magnet Emulation
	RFID reader/writer dengan antarmuka UART RS-232.

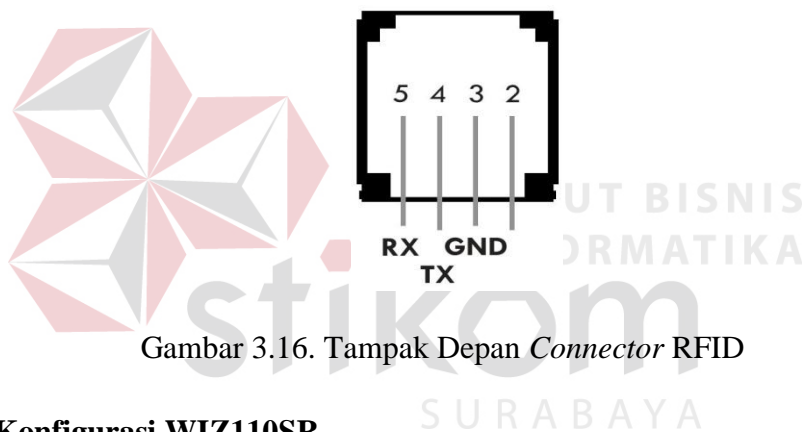
Gambar 3.15. Konfigurasi *Jumper* Pada Komponen *Reader* RFID

Adapun hubungan antara komputer dengan RFID Starter Kit adalah “*Straight*” dengan konfigurasi yang terdapat pada Tabel 3.3.

Tabel 3.3. Konfigurasi Hubungan Antara Komputer Dengan RFID

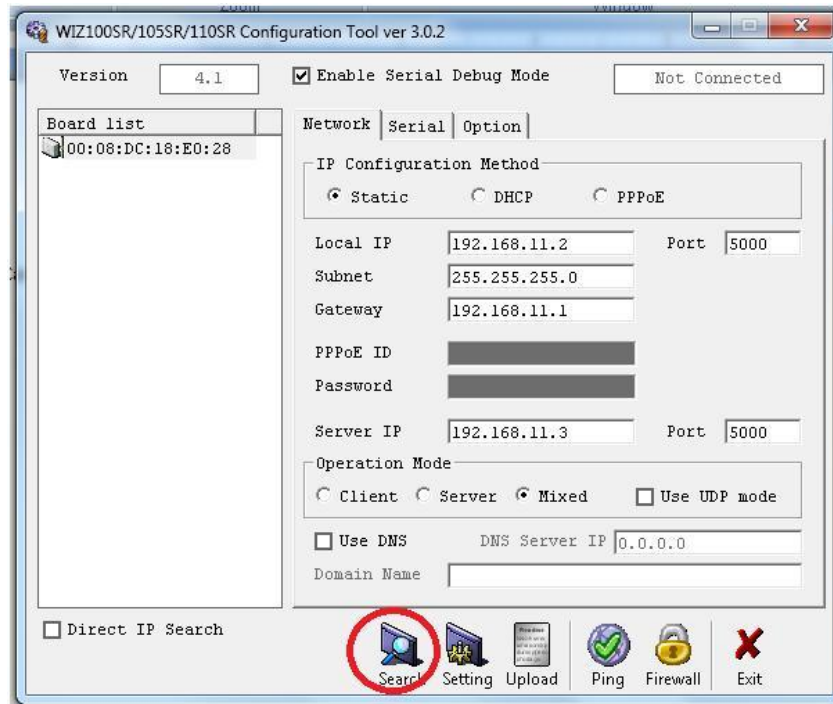
COM Port Komputer DB9	RFID Connector (J1)
RX (pin 2)	RX (pin 5)
TX (pin 3)	TX (pin 4)
GND (pin 5)	GND (pin 3)

J1 hanya digunakan untuk RFID *reader only* dalam *mode* UART RS-232 (ASCII) dan RFID *reader/writer*. Pada *mode* lain, J1 tidak boleh digunakan dan kabel tidak boleh terhubung. Pada Gambar 3.16 terlihat tampak depan *connector* serial yang didesain seperti RJ11.

Gambar 3.16. Tampak Depan *Connector* RFID

3.3.6. Konfigurasi WIZ110SR

Microcontroller dapat berkomunikasi melalui jaringan berbasis *internet protocol* menggunakan modul WIZ110SR, untuk itu diperlukan beberapa pengaturan pada modul WIZ110SR. Pengaturan tersebut dapat dilakukan melalui WIZ110SR *Configuration Tool*. Konfigurasi ini digunakan untuk merubah komunikasi serial ke komunikasi TCP/IP supaya *microcontroller* dapat berkomunikasi dengan PC *server*. Tampilan jendela pengaturan modul WIZ110SR dapat dilihat pada Gambar 3.17.



Gambar 3.17. Konfigurasi WIZ110SR

Pada Gambar 3.17 dijelaskan bahwa pertama kali pada konfigurasi WIZ110SR yaitu kita pilih tombol *search* supaya kotak *box* pada konfigurasi keluar IP *default* dari modul beserta informasi lainnya seperti versi *firmware* dari modul dan *mac address*.

Langkah-langkah keseluruhan dari pengaturan modul *ethernet* WIZ110SR adalah sebagai berikut :

1. Modul WIZ110SR dikoneksikan dengan komputer yang akan digunakan untuk proses konfigurasi melalui *network switch*.
2. Konfigurasi modul dilakukan dengan menggunakan WIZ110SR *configuration tool* seperti pada Gambar 3.17.
3. Memulai proses konfigurasi tekan tombol *search* pada *tool* untuk menampilkan daftar modul yang terkoneksi ke jaringan. Daftar modul akan tampil di sebelah kiri (*Board List*) pada Gambar 3.17.

4. Pilih salah satu *board* yang akan dikonfigurasi. Ketika dipilih, pada bagian kanan akan muncul konfigurasi yang telah disimpan ke dalam modul sebelumnya.
5. Pada *tool* ini terdapat 2 tab yang wajib dikonfigurasi. Masing-masing tab tersebut memiliki fungsi sebagai berikut :

a. Network

Mengkonfigurasi modul WIZ110SR terkait dengan bagaimana modul tersebut dapat berkomunikasi melalui jaringan, seperti *IP Address*, *Subnet Mask*, *Gateway*, dan *Port*. Pada tab ini, beberapa hal yang dapat dikonfigurasi adalah sebagai berikut:

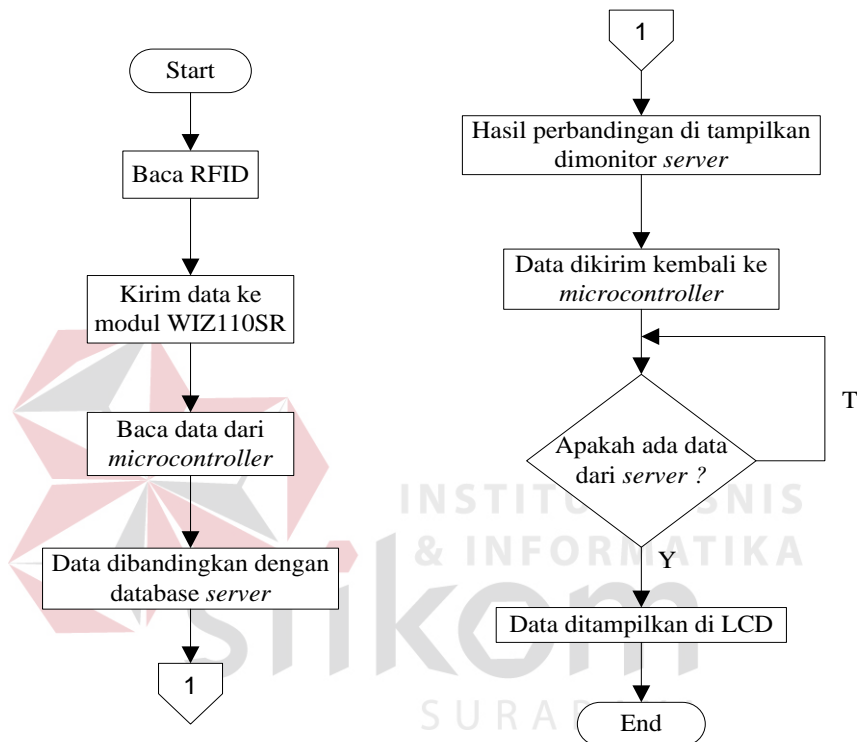
1. *IP Configuration Method*, digunakan untuk menentukan pengaturan alamat IP. Pengaturan alamat IP yang digunakan yaitu menggunakan *static IP*.
2. *Operation Mode*, digunakan untuk menentukan *mode* operasi dari modul WIZ110SR. *Mode* yang digunakan adalah *mode client*.

b. Serial

Mengkonfigurasi modul terkait dengan bagaimana modul dapat berkomunikasi dengan *microcontroller* melalui *Universal Asynchronous Receiver Transmitter* (UART) seperti *Baud Rate (Speed)*, Jumlah bit data setiap paket (*DataBit*), *Parity*, *Stop Bit*, dan *Flow Control*. Setelah semua terkonfigurasi sesuai (*Network & Serial*) tekan tombol pengaturan untuk mengirimkan konfigurasi ke modul WIZ110SR.

3.4. Perancangan Perangkat Lunak (*Software*)

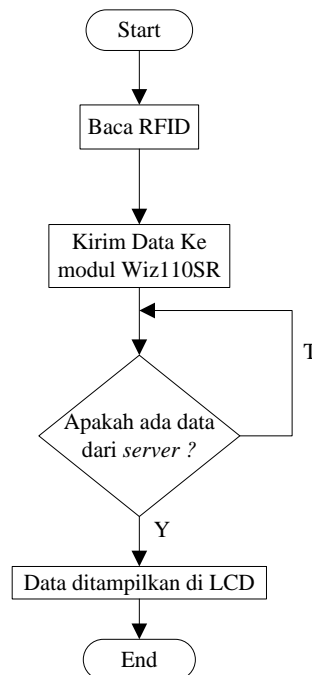
Perancangan program secara keseluruhan dibagi menjadi tiga bagian utama yaitu perancangan program *microcontroller*, perancangan aplikasi data absensi mahasiswa, dan program aplikasi komputer, Perancangan secara keseluruhan bisa dilihat lebih jelas melalui *flowchart* pada Gambar 3.18.



Gambar 3.18 *Flowchart* Keseluruhan Sistem

3.4.1. Perancangan Program *Microcontroller*

Microcontroller digunakan untuk mengolah data dari RFID lalu dikirimkan ke modul WIZ110SR melalui komunikasi serial. Perancangan program *microcontroller* secara keseluruhan bisa dilihat lebih jelas melalui *flowchart* pada Gambar 3.19.



Gambar 3.19. Flowchart Program Microcontroller

Dari Gambar 3.19 dijelaskan bahwa *tag* mengirimkan data ke RFID *reader*. RFID *reader* membaca kode dari *tag*. Kode tersebut dikirimkan ke *microcontroller* kemudian dilanjutkan ke PC melalui WIZ110SR. Data diolah di PC *server* untuk dibandingkan sesuai dengan database. Jika data tersebut belum diterima maka *microcontroller* menunggu. Jika data tersebut sudah diterima sesuai database maka data tersebut dikirimkan ke *microcontroller* dan data ditampilkan di LCD. Data yang ditampilkan tersebut berupa NIM, keterangan keterlambatan dan jam *display* RTC. Mereset seluruh program atau proses tersebut digunakanlah fasilitas *watchdog timer*.

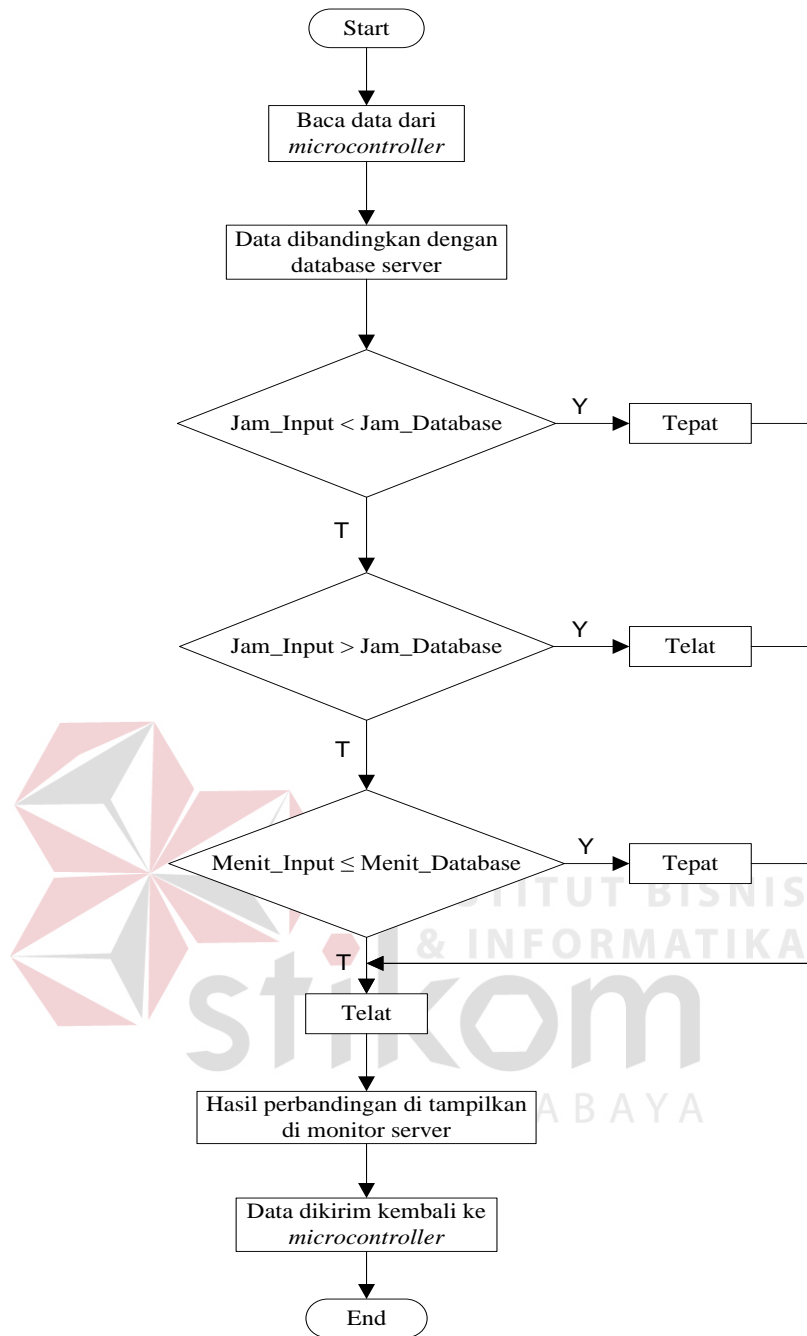
3.5. Perancangan Aplikasi Data Absensi Mahasiswa

Aplikasi data absensi mahasiswa dibuat menggunakan Visual Basic 6.0 sebagai antarmuka *software* dengan alat. Aturan penulisan program Visual Basic 6.0 didasari dengan aturan penulisan bahasa basic. Aplikasi komputer ini

digunakan untuk mengolah semua data yang dikirim dari *microcontroller*. Data dikirim secara serial ke komputer. Aplikasi ini terdiri dari 3 bagian yaitu Aplikasi *server*, aplikasi *filter*, dan aplikasi simulasi. Aplikasi *server* ini digunakan untuk menerima data *tag* ID dari *microcontroller* dan mengolah data ID menjadi data NIM sesuai database. Data yang telah dirubah tersebut ditambahkan catatan waktu keterlambatan kemudian data dikirimkan kembali ke *microcontroller* berupa NIM dan keterlambatan . Aplikasi *filter* digunakan untuk menyaring data yang ingin dicari berupa NIM, Kelas, dan Matakuliah supaya lebih ringkas dari data *form server*. Aplikasi simulasi digunakan untuk mensimulasikan alat absensi berupa *software* simulasi pada komputer, hal ini dikarenakan keterbatasan *hardware* yang dibuat hanya satu buah.

3.5.1. Aplikasi Server

Aplikasi *server* dipermudah dengan penjelasan *flowchart* pada Gambar 3.20. Penjelasan *flowchart* pada Gambar 3.20 bahwa program visual basic 6.0 menerima data dari *microcontroller* berupa kumpulan angka-angka yang digunakan sebagai ID dari *tag*. Kemudian data yang masuk tersebut dibandingkan dengan data yang ada di database dan ditampilkan. Setelah memperoleh hasil dari perbandingan tersebut data berupa NIM dan keterangan keterlambatan dikirim ke *microcontroller*.



Gambar 3.20. *Flowchart Program Server*

Data yang diinputkan berupa ID dari *microcontoller* kemudian ID tersebut diubah menjadi NIM. Data yang masuk disimpan dan dibandingkan dengan data yang ada di database kemudian dihasilkan sebuah keterangan keterlambatan. NIM

dan keterangan tersebut dikirimkan kembali ke *microcontroller*. Berikut ini contoh tabel data pada *server*.

Tabel 3.4. Contoh Data Pada *Server*

ID	NIM	NAMA	MATAKULIAH	JAM	MENIT
ADMIN	ADMIN	ADMIN	ADMIN	99	99
04007034FDBD	08410200056	Joko	Kalkulus 1	17	30
4D00D5875B44	08410200034	Jaki	PLC	7	30
0400801330A7	08410200004	Juki	Kalkulus 1	23	15

Data pada Tabel 3.4 ini yang nantinya akan digunakan pada pengujian Bab selanjutnya yaitu pada BAB 4. Desain dan kegunaan *form server* dapat dilihat pada Gambar 3.21.

Gambar 3.21. Desain *Form Server*

Keterangan Gambar 3.21:

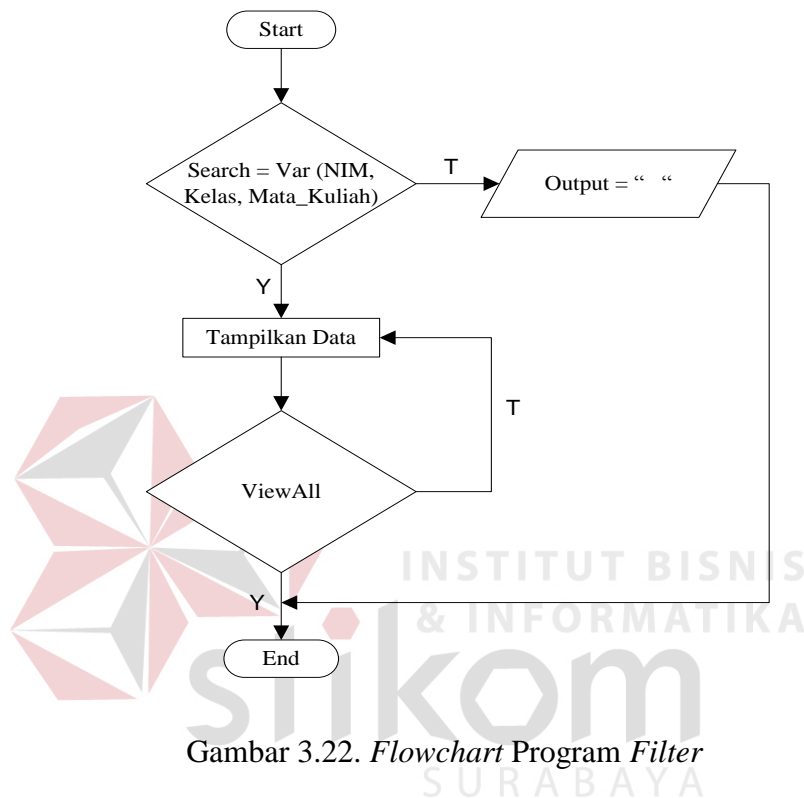
- ID : Menampilkan ID sesuai di *tag* (kartu) RFID.
- NIM : Menampilkan NIM mahasiswa.
- NAMA : Menampilkan nama mahasiswa.
- KELAS : Menampilkan kelas mahasiswa.

- MATAKULIAH : Menampilkan matakuliah mahasiswa.
- JAM : Menampilkan jam akan masuk proses belajar mengajar.
- MENIT : Menampilkan menit akan masuk proses belajar mengajar.
- MASUK : Menampilkan jam masuk yang setelah *tag* RFID pertama terdeteksi.
- KELUAR : Menampilkan jam Keluar yang setelah *tag* RFID kedua terdeteksi.
- JAM_SELESAI : Menampilkan jam selesai setelah jam belajar mengajar selesai.
- STATUS KELAS : Menampilkan keterangan terlambat atau tepat.
- STATUS ID : Pengecekan ID yang sudah terdaftar.
- KETERANGAN : Menampilkan simbol keterangan (T) Terlambat dan (P) Tepat.
- PORT : Menampilkan Port yang dipakai.
- CONNECT IP : Menampilkan koneksi IP yang sudah terhubung dengan *server*.
- SEARCH ID : Menampilkan pencarian ID yang diinginkan.
- Tombol Filter : Menampilkan *form filter*.
- Tombol New : Penambahan isi database.
- Tombol Save : Penyimpanan isi database.
- Tombol Delete : Penghapusan isi database.
- Tombol Find : Pencarian ID yang telah dimasukkan di Search ID.
- Tombol Exit : Keluar dari seluruh aplikasi.

- Kotak Merah : Hasil pembacaan data dari RFID yang terkirim dari *microcontroller*.

3.5.2. Aplikasi *Filter*

Aplikasi *filter* dipermudah dengan penjelasan *flowchart* pada Gambar 3.22.



Gambar 3.22. *Flowchart* Program *Filter*

Pada Gambar 3.22 menjelaskan bahwa setelah data yang ingin dicari ditulis pada kotak *box find* maka program *filter* akan menampilkan data sesuai dengan yang dicari (NIM, Kelas, dan Matakuliah). Jika tidak ada maka program filter akan menampilkan program kosong. Desain dan kegunaan *form filter* dapat dilihat pada Gambar 3.24.

Gambar 3.23. Form Filter

Keterangan Gambar 3.23 :

- Pilihan menu Filter : Pemilihan 3 menu yaitu NIM, kelas, dan matakuliah sehingga proses penyaringan dapat dilakukan dengan mudah.
- Text Filter : Pengisian sesuai dengan Pilihan menu *filter* yaitu NIM, kelas, dan matakuliah.
- Tombol Find : Pencarian sesuai dengan pilihan yang terdapat di Pilihan menu *filter* dan text *filter*.
- Tombol View All : Menampilkan semua data yang sama dengan data *server*.
- Tombol Exit : Hanya keluar dari aplikasi *filter*.
- Kotak Merah : Hasil penyaringan data yang ada pada *server*.

3.5.3 Aplikasi Simulasi

Pada aplikasi simulasi hanya untuk pengujian Tugas Akhir sebagai pengganti alat absensi (simulasi). Desain dan kegunaan aplikasi simulasi dilihat pada Gambar 3.24.



Gambar 3.24. *Form Simulasi*

Keterangan Gambar 3.24:

- Host IP : Menuliskan IP ke alamat *server* yang dituju.
- Port : Menampilkan Port yang ditujukan ke *server*.
- Tombol Connect : Menghubungkan koneksi antara aplikasi simulasi dengan aplikasi *server*.
- Tombol Disconnect : Memutuskan koneksi antara aplikasi simulasi dengan aplikasi *server*.
- Tombol Exit : Keluar dari aplikasi.
- Tombol Send : Mengirimkan isi dari text *send*.
- Text Send : Mengisi ID yang akan dikirim ke *server*.

3.6. Program Aplikasi Komputer

Pembuatan program aplikasi komputer menggunakan pemrograman visual basic 6. Penjelasan mengenai program dibagi menjadi tiga yaitu program *form server*, program *form koneksi (load)*, dan program *form TCP/IP*.

Berikut ini merupakan penjelasan mengenai program *form server*. *Form server* digunakan untuk menampilkan data absensi dari database. Sebelum menggunakan aplikasi ini, pengguna harus melakukan konfigurasi awal yaitu mengkonfigurasi koneksi, dan port.

Pada setiap pemrograman dibutuhkan beberapa variabel untuk membantu dalam pembuatan aplikasi. Berikut ini merupakan deklarasi variabel pada *form server*.

```
Dim a, b, c As Integer
Dim i(16) As String
Dim status As Boolean
Dim sHari As String
Dim aHari
Dim i As Long
Dim j As Long
Dim newWinsock As Boolean
Dim strData As String
```

Form load merupakan program yang akan dijalankan disaat *form* pertama kali dibuka. Pada *form load* secara garis besar berisi mengenai nilai awal dan sebagai proses inisialisasi seluruh *object* yang dibutuhkan.

```
Private Sub Form_Load()
    aHari = Array("Minggu", "Senin", "Selasa", "Rabu", _
        "Kamis", "Jumat", "Sabtu")
    Timer2.Interval = 500 'Set property interval
    Timer2.Enabled = True 'Aktifkan jika belum...

    With Winsock1(0)
        .Close
        .LocalPort = Text4.Text
        .Listen
    End With
```

Proses penerimaan data dari *microcontroller*. Proses ini menggunakan *winsock* sebagai *socket* (penghubung) untuk komunikasi TCP/IP. *Syntaxnya* berikut.

```
c = Index
Winsock1(Index).GetData strData
Text6.Text = Text6.Text & strData
```

Proses data ID masuk dibandingkan dengan database. Proses ini *update* secara terus-menerus dengan menggunakan *timer*. Berikut ini *syntaxnya*.

```
Private Sub Timer1_Timer()
If Text6.Text <> "" Then
Adodc1.Recordset.Find "id = '" + Text6.Text + "'", ,
adSearchForward, 1
If Not Adodc1.Recordset.EOF Then ' berulang sampai EOF
DataGrid1.Refresh
status = True
If Text12.Text <> "0" Then ' data tidak sama dg 0
Text13.Text = Format(Time, "hh:mm:ss")

Else
Text12.Text = Format(Time, "hh:mm:ss")
End If

If Cint(Text7.Text) = Cint(Format(Time, "h")) And
Cint(Text10.Text) < Cint(Format(Time, "n")) Then
If Text15.Text = "0" Then
Text15.Text = "TERLAMBAT"
Text9.Text = "(P)"
End If
ElseIf Cint(Text7.Text) = Cint(Format(Time, "h")) And
Cint(Text10.Text) >= Cint(Format(Time, "n")) Then
If Text15.Text = "0" Then
Text15.Text = "TEPAT"
Text9.Text = "(T)"
End If

ElseIf Cint(Text7.Text) > Cint(Format(Time, "h")) Then
If Text15.Text = "0" Then
Text15.Text = "TEPAT"
Text9.Text = "(P)"
End If
ElseIf Cint(Text7.Text) < Cint(Format(Time, "h")) Then
If Text15.Text = "0" Then
Text15.Text = "TERLAMBAT"
Text9.Text = "(T)"
End If
End If
Else
MsgBox "Maaf Datanya Kosong...."
Text6.Text = ""
End If
```

Setelah data ID dibandingkan dengan database maka data tersebut dikirim kembali ke *microcontroller* dengan menggunakan *winsock* . Berikut ini *syntax* pengiriman datanya.

```
If status = True Then
Text8.Text = "*" + Text2.Text + Text9.Text + "#"
b = 1
a = Len(Text8.Text) ' jumlah karakter text8

While (a >= b)
i(b) = Mid$(Text8.Text, b, 1) 'data yg ke berapa
Winsock1(c).SendData i(b) ' ngirim data array
b = b + 1
Wend
```

Port TCP/IP digunakan sebagai alat pada visual basic untuk dapat mengakses port TCP/IP pada komputer menggunakan *winsock*. Berikut ini merupakan *syntax* untuk menghubungkan koneksi antara alat dengan komputer pada port TCP/IP.

```
On Error GoTo errHandle

If Index = 0 Then
For i = 1 To Winsock1.UBound
If (Winsock1(i).State = sckClosed) Or (Winsock1(i).State =
sckClosing) Then
j = i Exit For
End If
Next i

If j = 0 Then
Call Load(Winsock1(Winsock1.UBound + 1))
j = Winsock1.UBound
newWinsock = True
End If

With Winsock1(j)
Call .Close
Call .Accept(requestID)
End With

If newWinsock Then
List1.AddItem Winsock1(j).RemoteHostIP & " [CONNECTED]"
Else
List1.List(j - 1) = Winsock1(j).RemoteHostIP & " [CONNECTED]"
End If
End If

Exit Sub
errHandle:
Call Winsock1(0).Close
```

Setelah data sudah masuk semua pada aplikasi *server* maka data dapat di *filter* dengan menekan tombol *filter* pada aplikasi *server*. *Syntax filter* pada tombol *find* untuk mencari NIM, MATAKULIAH, dan KELAS sebagai berikut.

```
Private Sub Command1_Click()
If txtFilter.Text = "" Then
Adodc1.ConnectionString = "DSN=alex"
Adodc1.RecordSource = "alex"
Adodc1.Refresh
Adodc1.Recordset.Filter = ""
Else
```

Syntax pencarian NIM sebagai berikut.

```
If CBFILTER.Text = "NIM" Then
Adodc1.ConnectionString = "DSN=alex"
Adodc1.RecordSource = "alex"
Adodc1.Refresh
Adodc1.Recordset.Filter = "NIM like '%" + txtFilter.Text + "%'"
```

Syntax pencarian KELAS sebagai berikut.

```
ElseIf CBFILTER.Text = "KELAS" Then
Adodc1.ConnectionString = "DSN=alex"
Adodc1.RecordSource = "alex"
Adodc1.Refresh
Adodc1.Recordset.Filter = "KELAS like '%" + txtFilter.Text + "%'"
Else
```

Syntax pencarian MATAKULIAH sebagai berikut.

```
Adodc1.ConnectionString = "DSN=alex"
Adodc1.RecordSource = "alex"
Adodc1.Refresh
Adodc1.Recordset.Filter = "MATAKULIAH like '%" + txtFilter.Text + "%'"
End If
End If
End Sub
```

Syntax tombol *viewall* untuk mengembalikan data seperti pada aplikasi *server* sebagai berikut.

```
Private Sub Command2_Click()
txtFilter.Text = ""
Adodc1.ConnectionString = "DSN=alex"
Adodc1.RecordSource = "alex"
Adodc1.Refresh
Adodc1.Recordset.Filter = ""
End Sub
```